Multimedia Systems (2017) 23:85–93
DOI 10.1007/s00530-014-0429-2

SPECIAL ISSUE PAPER

A human motion feature based on semi-supervised learning of GMM

Tian Qi · Yinfu Feng · Jun Xiao · Hanzhi Zhang · Yueting Zhuang · Xiaosong Yang · Jianjun Zhang

Published online: 21 October 2014

Abstract Using motion capture to create naturally looking motion sequences for virtual character animation has become a standard procedure in the games and visual effects industry. With the fast growth of motion data, the task of automatically annotating new motions is gaining an importance. In this paper, we present a novel statistic feature to represent each motion according to the pre-labeled categories of key-poses. A probabilistic model is trained with semi-supervised learning of the Gaussian mixture model (GMM). Each pose in a given motion could then be described by a feature vector of a series of probabilities by GMM. A motion feature descriptor is proposed based on the statistics of all pose features. The experimental results and comparison with existing work show that our method performs more accurately and efficiently in motion retrieval and annotation.

Keywords Human motion feature · Semi-supervised learning · Probabilistic model · Motion retrieval · Motion classification

1 Introduction

The growing popularity of motion capture (mocap) technique in feature films and interactive entertainments has led to an explosive growth of motion data. The reuse of those motion data to create realistic motions for new characters by applying motion editing [1, 2], motion synthesis [3–5] and motion retargeting [6] techniques has become the focus of research in the past several years. However, prior to reusing and processing the old motions, one fundamental problem of identifying and extracting similar motion clips from the database has to be solved. It is essentially a motion matching problem. The general procedure is to calculate a concise and representative feature for each motion, and then compare the similarities with all other motions in the mocap database. The efficiency and accuracy of these motion retrieval and annotation processes largely depend on the property of the used features.

Most early work [7] uses textual description such as running, walking, to label existing motions in a database. It does not only involve a lot of manual work, the textual label is also too short and general to fully represent the features of each motion. Later works [8–11] use the numeric-based features and logic-based features involving the 3D coordinates of each joint in each frame. It includes too much redundant information and the 'huge' feature makes the motion matching really slow. Some recent works [12] present semantic features which better represent the essence of motions and the low dimension of features largely speeds up the motion retrieval process. In this paper, we present a new feature in this category. The work [13] has shown that a human motion clip could be described with some representative poses, which we call the 'key-pose'. Intuitively, two similar motions may share most key-poses, while the motions belonging to different motion classes may share none or only a few key-poses (as shown in Fig. 1). A good selection of key-poses can be used to represent different motion classes. The second benefit to use key-poses as a feature is, although the category of motions is infinite, the types of key-poses are relatively limited. A new category

T. Qi · Y. Feng · J. Xiao (✉) · H. Zhang · Y. Zhuang
Institute of Intelligence Artificial, Zhejiang University,
38 Zheda Rd, Hangzhou 310027, Zhejiang, China
e-mail: junx@cs.zju.edu.cn

X. Yang · J. Zhang
National Centre of Computer Animation, Bournemouth
University, Poole, Dorset BH12 5BB, UK

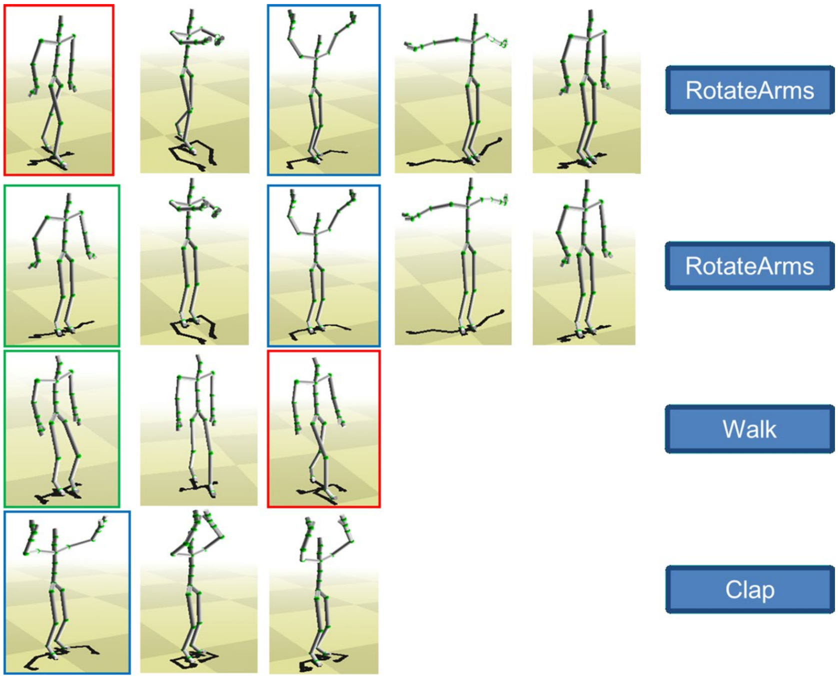

Fig. 1 Examples of key-poses in four motions. The *top two motions* are from the same motion class (*RotateArms*) and they share almost all key-poses. However, the *third motion* (*Walk*) and the *last motion* (*Clap*) share only a few key-poses with the top two. The *red, green* and *blue rectangles* represent the only three pairs of similar key-poses that shared between different kinds of motions (one color for one pair)

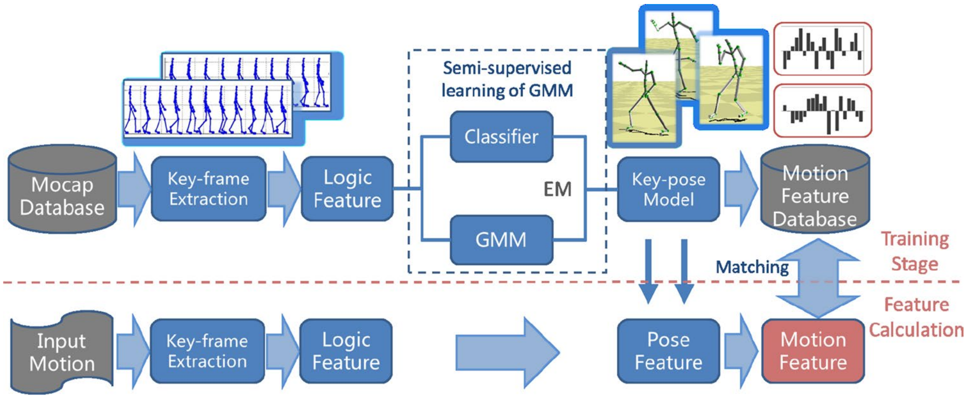

Fig. 2 The flowchart of our approach

of motions could be composed of existing key-poses. But unfortunately, most state-of-the-art research classifies a motion by its pose feature directly, and does not utilize the 'key-pose' as a middle level between the pose feature and the motion class. In the work of Qi et al. [12], the Gaussian mixture model (GMM) is applied to model the key-poses of every motion class. However, the result of this unsupervised learning method shows that the mean value of each Gaussian model may not align with the motions' semantic category. To improve the result, we manually define a class of key-poses from close observation of all the motions in our database, and partially label some poses for semi-supervised learning.

In this paper, we present a novel probabilistic human motion feature and a semi-supervised learning of GMM method to train the key-pose model. The flowchart of our work is shown in Fig. 2. First, some key-frames are extracted for each motion in the database. Then a logic-based feature called General Position Feature (GPF) is extracted on each key-frame. Since the key-frames are partially labeled according to the defined key-pose categories, we construct both the classifier (supervised learning) and the GM model (unsupervised learning). Similar to [14], a combined semi-supervised learning method based on Expectation-Maximization (EM) algorithm is introduced to model the key-poses by a set of GM parameters. Then given an unknown input motion clip, a probabilistic pose feature is calculated for each pose by the key-pose model. Finally, the motion feature is generated by combining the statistical value of all the poses in the motion clip.

As the main contribution of this paper, we propose a novel probabilistic human motion feature based on semi-supervised learning of GMM. Unlike the other state-of-the-art research, the key-frames of our database are partially labeled by a series of specified key-poses, which could be well estimated by our feature model. Therefore, the complete motion matching process is divided into two parts, pose recognition and motion matching, which is close to human perception. In addition, our probabilistic model contains more information than general clustering methods, and the semi-supervised learning method is able to prevent overfitting (both of which will be discussed in Sect. 3 in detail).

2 Related work

Multimedia content analysis and understanding is a crucial research problem, where designing a discriminative feature is a basic way to improve the performance. In recent work, both features and models have been worked out for many applications in multimedia area, such as image recognition [15], retrieval [16], cropping [17], segmentation [18, 19], and video annotation [20]. Unfortunately, in our 3D human motion applications, there is still lack of an efficient feature for motion representation.

With the rapid growth of mocap databases, the applications based on large motion data repository, such as motion retrieval and data-driven motion recognition or annotation, become popular, and a lot of research has been focused on them in recent years. As described in the last section, motion matching, the key procedure, could be divided into two parts, extracting pose features and comparing the similarity. They rely on the answers of two further issues, which are how to construct a concise and representative feature, and how to compare motions of different length by pose features.

In the past two decades, a great deal of research has been carried out for skeleton feature extraction, which could be concluded as two categories, the numeric-based features and the logic-based features. A numeric-based feature is obtained from the original data directly, regardless of its physical meaning, while a logic-based feature concerns more about the joint relationship in a real skeleton hierarchy structure. With the increasingly precision of mocap devices, the dimensionality of mocap data gets bigger. Principle Component Analysis (PCA) is applied [21–23] to reduce the dimensionality of poses in motion. For further improvement, Forbes et al. [8] employ the weighted PCA to distinguish the different importance of different skeleton nodes simultaneously. In addition, some signal processing methods, such as wavelet transform [9], are also introduced. As shown in Fig. 1, motions could be represented by a series of key-poses. Therefore, all poses in a database could be clustered, and the idea of using motion clustering indices (MCI) to represent poses is adopted by [21, 24, 25]. However, all numeric-based features cannot describe the logic meaning of a motion, e.g. the relative locations of skeleton joints. It is a fatal weakness of this kind of feature, as logically similar motions may not be numerically similar [26].

Therefore, a lot of logic-based features are presented to describe motion in recent research. Muller et al. proposed a Boolean feature that describes geometric relations between specified points of a pose [10], e.g. the right hand is in front or behind the body plane. Such a series of Boolean values are combined to describe each pose in a motion. This work is extended by Chen et al. [11] In their work, 10 types of relational geometric features (RGF) are defined, which use the basic elements of points, lines and planes to calculate the relative position of joints in each pose. Similarly, Tang et al. [27] take the relative distance between the joints as the basic features, to calculate the similarity of a pair of motions. The feature is improved to retrieve logically relevant motions by [28]. Those geometric features are also applied and extended by [29], where a combined relational geometric feature of over 20,000 dimensions is used. Since the feature is high dimensional, feature selection methods such as Adaboost are usually needed.

However, both low-level numeric and logic-based features could only describe or represent a pose. They cannot make use of semantic pose labels as a part of the feature directly. As discussed in the last section, the specified pose labels can be used to improve the accuracy of recognition only when the key-poses in a motion are recognized. So in this paper, we still use the logic-based features, but present a high-level probabilistic feature based on semi-supervised learning of GMM, which utilizes both the logic information and the labels of a pose.

Once the skeleton features are extracted on pose, another difficult point is how to compare motions with a different length by pose features. The dynamic time warping (DTW) algorithm is most widely used here, due to its effectiveness [8, 29, 30]. It is a dynamic programming (DP)-based algorithm that finds a path with a minimal distance between two motions. In addition, some string matching methods could be applied if a pose in a motion is represented as a character (such as MCI). In the work of [25], human motions are broken into hierarchical parts, and clustered by k-means. The motion clustering indices (MCI) are used to represent each motion pattern. A classical Knuth–Morris–Pratt (KMP) string matching algorithm is applied and extended to compare motions. Qi et al. [31] generate the 'action string' from a motion, and a string matching algorithm based on dynamic programming (DP) is used in motion matching.

However, the computational cost of either DTW or those string matching algorithms is very high, which cannot meet the real-time requirement of most applications, especially for interactive applications. But if a pose feature is probabilistic, there is a way to prevent this disadvantage, that is

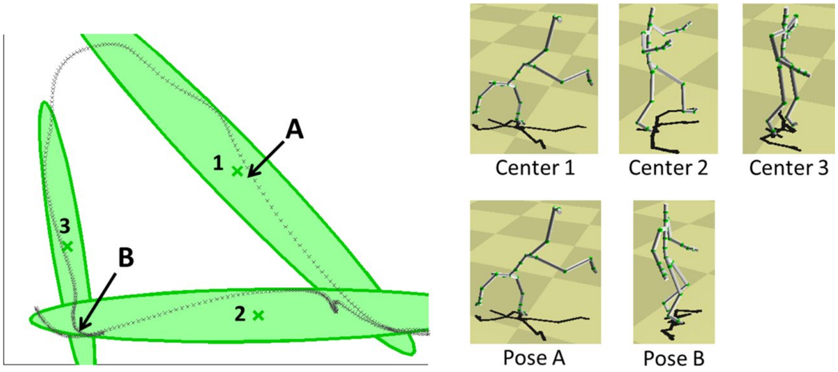

Fig. 3 The data distribution of a real motion with the category of 'cartwheel'. The three large *x-marks* represent the center locations of key-poses generated by unsupervised learning algorithms. In this scheme, pose *A* could be well represented, but pose *B* cannot. The three clustering centers, poses *A* and *B* are visualized in the right

taking the normalized statistical value of each pose feature as the descriptor of the motion [12]. The computation is very fast, and the performance is robust in motion matching. This idea is adopted in this paper, so the feature model is trained by the probabilistic model GMM. However, in the work of [12], the key-poses are estimated separately in each motion class, which will cause some redundancy, because the common key-poses shared in different motion classes are duplicated in different models. In this paper, a semi-supervised learning method is applied to take full advantage of the partial pose labels, and all the key-poses are estimated only once in the same time, to overcome the redundancy.

3 Motivation

In this section, we discuss the detail and the justification on why we choose the GMM rather than the K-means to model the key-poses, and why we choose semi-supervised learning rather than unsupervised or supervised learning, which is a key idea of this paper.

3.1 GMM vs. K-means

As described in Sect. 1, key-poses can be chosen to represent all kinds of motions. A simple way to estimate key-poses is using the K-means algorithm to build clusters, in which each pose is labeled by the nearest clustering index. However, if there is a transitional pose between two neighbor key-poses, it must be labeled as one of them, which is not always necessary. Unlike the k-means, the GMM is able to allow soft assignments by providing a probability for a given pose that belongs to a cluster.

For example, Fig. 3 shows the data points of poses in a real motion with the category of 'cartwheel', which are projected to a 2D space with the first two principle components. The three large x-marks represent the center locations of key-poses generated by unsupervised learning algorithms. If K-means is applied, although the pose of point *A* is perfectly labeled to class 1, the transitional pose of point *B* has to be forced to class 2 or 3. Unlike K-means, when GM model is used (the ellipse of each center represents the covariance of that Gaussian), the descriptor of point *A* is the same, but the *B*'s descriptor could be a set of probabilities as $\{0, 0.5, 0.5\}$, which could well describe this pose.

3.2 Unsupervised vs. supervised learning

Although the mean of each GM model could estimate the key-pose, it may not be the right position we expected, because the learning is unsupervised. If the motion is well labeled for each pose, the key-pose could be pointed out by calculating the average positions of all poses with the same corresponding label. Figure 4 gives the same example as Fig. 3. The raw data points are distributed as Fig. 4a, the three different marks(x-marks, circle marks and star marks) represent the three different pose labels on each frame. The subfigure 4b shows the expected key-poses (large red circle marks) and the estimated positions (green ellipses with blue x-marks as their centers) by unsupervised learning of GMM. When supervised and semi-supervised learning of GMM [14] is applied to take advantage of the pose labels, the estimation result is much closer to the expected positions (shown in Fig. 4c, d).

However, since the key-poses are defined and labeled manually, it may be too specific to cause overfitting when taking the totally supervised learning algorithms. Because the semi-supervised learning algorithms utilize distribution of those unlabeled data to expand the training set, they are more likely to prevent overfitting.

On the other hand, it has to be full-labeled on the database, if a totally supervised learning method is applied. However, setting the pose label on a large mocap data repository costs a great deal of manual effort. Moreover, when a new category of motion is added into the database, the key-poses included may already exist in the original database. So it is not necessary to label any pose in the new motion category if a semi-supervised learning method

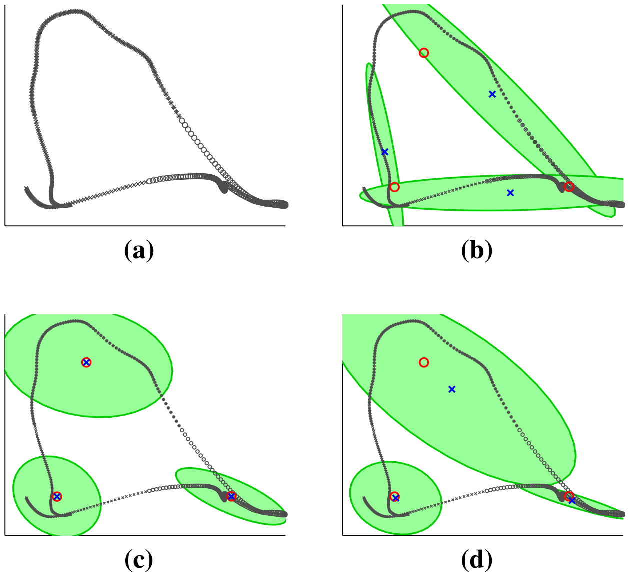

Fig. 4 The key-pose location estimation by unsupervised, supervised and semi-supervised learning algorithms. **a** The data distribution. **b** The unsupervised learning result (original GMM). **c** The result of supervised learning of GMM. **d** The semi-supervised learning result of GMM

is used. Actually, this approach has been already adapted in visual recognition area [32, 33], but rarely used in 3D human motion applications.

Following the above analysis, we apply an efficient semi-supervised learning method of the probabilistic GM model [14] to both get a better performance and save labor cost.

4 Human motion feature generation

4.1 Notations and pre-processing

Motion data consist of 3D joint positions frame by frame. A motion clip s can be represented as $s = \{f_1, f_2, \ldots, f_s\}$, where f_i contains the x, y, z coordinates of each joint in the i-th frame.

As mentioned before, logically similar motions may not be numerically similar. To describe the logic meaning of a motion, we present a simple but robust logic-based feature, called the GPF, which contains four parts. First, the velocity of each joint, which is calculated by its position offset from the last frame divided by the time interval. If there are K joints in the skeleton model, the data dimensionality of this part is K. Second, the acceleration of each joint, which is variation of velocity between the current and last frame, with K dimensions in each frame. Next, the relative distances, calculated between each pair of two arbitrary joints, with $K \times (K-1)/2$ dimensions. And last, the distance of each joint to the body center (root joint), with $K-1$ dimensions. The GPF, therefore, contains a total of $K \times (K-1)/2 + 3K - 1$ dimensions for each frame, and the feature in each part is normalized to avoid a bias. Since there is some redundancy in GPF, especially in the third part, we adopt PCA to reduce its dimensionality and find the principle feature subspace. The above four relational geometric features (RGFs) are chosen in our GPF feature, because they could well describe a pose, and is fast enough for online computation.

After the GPF feature extraction, we apply the K-means clustering-based key-frame extraction algorithm, introduced by [12], and then a set of key-frames are selected as $\hat{s} = \{x_1^T, x_2^T, \ldots, x_n^T\}^T$, which is an $n \times m$ matrix, where m is data dimension of the principle feature subspace, n is the number of key-frames and x_i is the feature data of i-th key-frame.

4.2 Semi-supervised learning of GMM

Assuming there are altogether N key-frames selected from all motions in a database, the specified P key-poses are learned by the semi-supervised learning algorithm of GMM. Each Gaussian model $\mathcal{N}(\mu_i, \Sigma_i), i = 1, 2, \ldots, G$, represents an estimated data distribution of one or more certain key-poses, and there are a total of G Gaussian models mixed together to describe all key-poses. Since the key-poses are specified and labeled manually, there will be errors. In detail, in a few cases, the manually labeled two separate categories of key-poses may be very similar actually, as well as the poses labeled to the same kind of key-poses may be varied a lot. Therefore the number of Gaussian G may not necessarily be equal to P in this scheme, to reduce the error caused by manual labels.

In a general GM model, the initial prior of each Gaussian is set to $p(q_i|\Theta) = 1/G$, where q_i is the i-th cluster, and Θ is the parameter set. And the posterior probability $p(q_i|x_k, \Theta), k = 1, 2, \ldots, N$ is calculated as follows:

$$p(q_i|x_k, \Theta) = \frac{p(q_i|\Theta) \cdot p(x_k|q_i, \theta_i)}{\sum_t p(q_t|\Theta) \cdot p(x_k|q_t, \theta_t)}. \tag{1}$$

Unlike the standard GM method, the semi-supervised GM algorithm [14] we adopted does not only take the advantage of probabilistic data distribution given by GM, but also utilizes the partial labels by a supervised classifier. First, a supervised classifier must be trained to give a suggestive label to unlabeled data, $p(c_j|x_k)$, where $c_j (j = 1, 2, \ldots, P)$ represents the j-th key-pose class. Then $p(x_k|c_j)$ is calculated by the Bayesian rule. According to that, a mapping from each cluster q_i to each key-pose class c_j could be estimated as:

$$p(c_j|q_i) = \frac{\sum_k p(x_k|c_j) \cdot p(x_k|q_i)}{\sum_t \sum_k p(x_k|c_t) \cdot p(x_k|q_i)}. \tag{2}$$

Then the supervised part of each cluster $F(q_i)$ is defined as probabilistic style:

$$F(q_i) = -\log\left(-\sum_j p(c_j|q_i)\cdot\log(p(c_j|q_i))) + \log(\log(P)\right). \tag{3}$$

To optimize both the supervised part and the unsupervised part (traditional GM model), the objective function, which contains the two-fold uncertainty $p(c_j|x_k)$ and $p(q_i|x_k)$ is defined as:

$$J(\Theta) = (1-\alpha)\sum_{x_k}\log(p(x_k|\Theta)) + \alpha\sum_i^G \log(F(q_i)), \tag{4}$$

where $\alpha(0 \le \alpha \le 1)$ is a balance parameter to determine the proportion between unsupervised and supervised learning. The optimal parameter set Θ^* is found by maximizing the objective function $\Theta^* = \operatorname{argmax}_\Theta J(\Theta)$.

The Expectation-Maximization algorithm is applied to solve the equation above iteratively. The μ_i, Σ_i and the weight $p(q_i|\Theta)$ are updated as follows:

$$\mu_i = \frac{\sum_k\{(1-\alpha)p(q_i|x_k) + \alpha B_i\sum_j a_i^j p(x_k|q_i)p(x_k|c_j)\}x_k}{\sum_k\{(1-\alpha)p(q_i|x_k) + \alpha B_i\sum_j a_i^j p(x_k|q_i)p(x_k|c_j)\}}, \tag{5}$$

$$\Sigma_i = \frac{\sum_k\{(1-\alpha)p(q_i|x_k) + \alpha B_i\sum_j a_i^j p(x_k|q_i)p(x_k|c_j)\}A_i^k}{\sum_k\{(1-\alpha)p(q_i|x_k) + \alpha B_i\sum_j a_i^j p(x_k|q_i)p(x_k|c_j)\}}, \tag{6}$$

$$p(q_i|\Theta) = \frac{\sum_k p(q_i|x_k)}{N}, \tag{7}$$

where $a_i^j = 1 + log(p(c_j|q_i))$, $A_i^k = (x_k - \mu_i)(x_k - \mu_i)^t$ and B_i is defined as:

$$B_i = \frac{-1}{F(q_i)[\sum_j p(c_j|q_i)\cdot log(p(c_j|q_i))]}. \tag{8}$$

The overall process of this algorithm (called GEMP)[14] in our application of key-pose model estimation is presented as below.

Input: The pre-processed dataset $X = \{x_1, \ldots, x_N\}$, and the partial corresponding specified label in P key-poses.

Output: The final GM parameters Θ: μ_i, Σ_i and weight $p(q_i|\Theta)$, $\forall i$.

Task:

(1) Set $t = 0$

(2) Train a classifier that can provide $p(c_j|x_k)$ for $\forall j, k$.

(3) Use K-means to find the initial parameters Θ_0.

(4) **E-step:** Calculate $p(x_k|q_i)$ by Θ_t, and estimate $p(q_i|x_k)$, $p(x_k|c_j)$, B_i and a_i^j, $\forall i, j, k$, by the equations above.

(5) **M-step:** Update parameters Θ_{t+1} using Eqs. 5, 6 and 7.

(6) Set $t = t + 1$

(7) Repeat (4) to (6) until convergence.

4.3 Motion feature generation

As the key-poses are described as a set of GM parameters Θ, our pose feature could be generated from the GM model. For a given motion s, an $n \times m$ feature matrix $\hat{s} = \{x_1^T, x_2^T, \ldots, x_n^T\}^T$ is obtained after the key-frame selection and GPF feature extraction, as represented in Sect. 4.1. For each pose $x_k, k = 1, 2, \ldots, n$, the probabilities $p(x_k|q_i)$ that x_k belonging to each clusters $q_i, i = 1, 2, \ldots, G$ is calculated by the GM parameter set Θ. Thus the pose feature $t^{(p)}$ for x_k is defined as:

$$t_k^{(p)} = \left\{\hat{p}(x_k|q_1), \hat{p}(x_k|q_2), \ldots, \hat{p}(x_k|q_n)\right\}, \tag{9}$$

where $\hat{p}(x_k|q_i)$ is normalized from $p(x_k|q_i)$, which subject to $\sum_{i=1}^G \hat{p}(x_k|q_i) = 1$.

Since our pose feature is probabilistic, it can easily describe a complete motion taking the normalized statistical value of each pose feature, which could avoid applying the time-consumed DTW algorithm. The motion clip feature could be described as:

$$t^{(m)} = \frac{1}{n} \times \left\{\sum_{k=1}^n \hat{p}(x_k|q_1), \sum_{k=1}^n \hat{p}(x_k|q_2), \ldots, \sum_{k=1}^n \hat{p}(x_k|q_n)\right\}. \tag{10}$$

5 Experiments

In this paper, the dataset we used for experiments is from HDM05 [34]. The well-segmented motion database contains 130 motion classes, but many of them are very similar (e.g. 'walk2StepsLstart' and 'walk2StepsRstart'). So we combine them into 24 basic motion classes, including 2,073 motion clips and over 420,000 frames. To take advantage of pose labeling, we defined a total of 82 key-pose classes, and the key-frames of about 20 % motions in each motion class are manually labeled to those key-pose classes. In our experiments below, half of the motions in dataset are served as training data, and the others are testing data.

Our method is implemented using MATLAB, and all experiments are executed on a computer with an Intel Core i5 3570 3.4 GHz and 8 GB of RAM.

5.1 Compared methods

In this paper, a semi-supervised learning of GMM is introduced to construct the key-pose model, where the key-poses are specified and poses in database are partially labeled manually. To prove the improved performance when taking

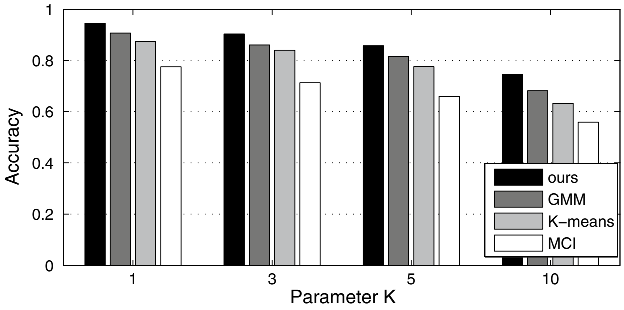

Fig. 5 The experimental result for motion retrieval. The methods of MCI, K-means, original GMM and ours are compared in each situation of $K = 1, 3, 5, 10$

advantage of pose labels, two unsupervised methods, the original GMM method and K-means clustering algorithm, are implemented. In addition, another method using MCI [21] is also executed for comparison purposes, where the same logic feature is extracted.

The parameters in the above algorithms are optimized for best performance. In our method, the two important parameters G and α are set to 75 and 0.5, and the Support vector machine (SVM) is chosen to be the classifier for the supervised part. While in GMM, K-means and MCI algorithms, the optimal value of the same parameter G, the number of pose clusters, are 55, 70 and 40, respectively.

5.2 Motion retrieval

Motion retrieval from a large data repository is a well-researched topic in recent years, where the key problem is the similarity calculation between two motions. In this subsection, the above four algorithms (MCI, K-means, GMM and ours) are compared. For each input motion, the first K most similar motions obtained by K-nearest-neighbor (KNN) are selected as the retrieval result. However, there is no general criterion to evaluate the retrieval results. In our experiments, if a retrieved motion belongs to the same motion class with the input motion, which means they are logically similar, it will be treated as a 'hit', otherwise a 'miss'. The accuracy of each algorithm with different situations $K = 1, 3, 5, 10$ is shown in Fig. 5.

Table 1 The detailed experimental results for motion classification

	Total	Ours	GMM	K-means	MCI	GMM + K-means	SLMC
Cartwheel	14	13	14	14	8	14	6
Clap	32	30	26	27	29	29	19
ElbowToKnee	40	40	40	40	40	40	35
Grab	110	107	96	100	86	100	45
HitHandHead	6	3	0	2	0	0	0
Hop	37	34	32	36	35	37	22
Hop1leg	69	64	65	65	61	66	41
Jog	32	32	31	30	30	32	10
JumpingJack	32	32	32	32	32	32	12
Kick	86	80	84	63	50	83	55
LieDownFloor	10	7	3	3	6	4	2
Punch	88	85	85	70	59	86	40
RotateArms	96	96	96	96	96	95	92
RunOnPlace	72	68	65	67	53	69	33
Shuffle	25	25	19	14	11	17	11
SitDown	38	28	33	28	29	31	13
Skier	20	20	20	20	16	20	15
Sneak	31	29	29	25	25	29	10
Squat	32	32	32	32	32	32	21
StandUp	48	37	33	33	33	35	13
Throw	14	11	14	8	3	11	2
ThrowBasketball	7	6	3	5	4	5	0
Walk	47	46	39	41	38	43	33
WalkBackward	15	15	8	7	15	9	3
WalkOnPlace	30	26	30	28	8	29	7
Overall (%)	100	93.7	90.1	85.9	77.5	91.9	52.4

The methods of original GMM, K-means, MCI, the combination of GMM and K-means (GMM + K-means), supervised learning of motion categories (SLMC) and ours are compared separately with different motion classes

Table 2 The comparison of time expended in each part

	Ours	GMM	K-means	MCI
GPF feature (training set)	154 s	154 s	154 s	154 s
Model construction	5,962 s	1,376 s	275 s	128 s
Clip feature (training set)	227 s	155 s	197 s	157 s
Training stage total	6,343 s	1,685 s	626 s	439 s
GPF feature (per clip / per frame)	161/0.6 ms	161/0.6 ms	161/0.6 ms	161/0.6 ms
Motion feature (per clip)	210 ms	143 ms	174 ms	53 ms
Motion retrieval (per clip)	2 ms	2 ms	2 ms	2,350 ms
Testing stage total (per clip)	373 ms	306 ms	337 ms	2,564 ms

Our method spent more time in the training stage, but less in the testing stage. The result shows our method meets the real-time requirement

The experimental results show that our method outperforms the other three in retrieval accuracy. The original GMM is weaker than ours, because it cannot take advantage of the pose labels, while K-means is weaker than original GMM,because the probabilistic feature takes more information than pure clustering method. MCI also takes clustering index to represent each pose, but the matching method of two sequences is not discriminative enough, so its performance is not better than others. Moreover, as DTW is used as the matching method, it is very time consuming in similarity calculation.

5.3 Motion classification

Motion classification is another important application. Similar to motion retrieval, in our experiments, the KNN is applied to search for the K retrieved motions. The motion class which contains most retrieved motions is chosen as the classification result. The classification accuracy of the above four algorithms is compared with the parameter $K = 4$. To give a more competitive comparison, we even combine both the GMM and K-means features and generate a high dimension representation (GMM+K-means). In addition, a supervised learning algorithm is added, where each pose is labeled by its motion category (SLMC), to prove the necessity of our manual labels. Table 1 shows the experimental results separated with different motion classes. Our method again outperforms the others for most motion classes, which proves the effectiveness of our motion feature.

The time expended on each part of our method is shown in Table 2, where the above three basic methods(MCI, K-means and GMM) are also compared. It can be concluded that our method is more time consuming on the training stage than the others, but our method is very fast in the testing stage. As the GPF feature could be extracted online, and the total time spent on motion matching is only 212 ms, the input motion data collected from real-time devices (such as Microsoft Kinect) could be recognized in real time.

6 Discussion and future works

In this paper, a novel probabilistic motion feature is presented, based on semi-supervised learning of GMM, which can well estimate the key-poses in human motions, and take full advantage of the manually specified pose labels. The experimental results show that our method outperforms the unsupervised learning methods (original GMM and K-means) and the state-of-the-art (MCI). The time consumed by our method in the testing stage is fast enough for real-time applications.

As a main disadvantage, our method takes the statistical probabilities of all pose features as the motion descriptor, so it cannot keep the timing sequence in a motion. But on the other hand, it saves much time on the motion matching process, which is a justifiable trade-off between efficiency and accuracy.

Our method mainly focuses on the feature extraction process, and the motion matching process could be optimized. Other than using the simple KNN algorithm to search the entire database, some data structure, such as K-D tree, could be applied to reduce the time complexity in the retrieval of similar motions, which is taken as one of our future works.

In addition, as our GMM-based motion feature is high dimensional, it would be beneficial to perform feature selection based on the newly designed feature. In our future works, we plan to introduce some research in state-of-the-art [33, 35] to have a more compact and discriminative representation.

Acknowledgments This research is supported by the National High Technology Research and Development Program (2012AA011502), the National Key Technology R&D Program (2013BAH59F00), the Zhejiang Provincial Natural Science Foundation of China (LY13F020001), the Fundamental Research Funds for the Central Universities (2014FZA5013), Zhejiang Province Public Technology Applied Research Projects (No. 2014C33090).

References

1. Gleicher, M.: Motion editing with spacetime constraints. In: Proceedings of the 1997 symposium on interactive 3D graphics. pp. 139–148 (1997)
2. Min, J., Liu, H., Chai, J.: Synthesis and editing of personalized stylistic human motion. In: Proceedings of the 2010 ACM

SIGGRAPH symposium on Interactive 3D Graphics and Games. pp. 39–46 (2010)

3. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. ACM Trans. Gr. (TOG) **21**(3), 473–482 (2002)
4. Jain, S., Ye, Y., Liu, C.: Optimization-based interactive motion synthesis. ACM Trans. Gr. (TOG) **28**(1), 10 (2009)
5. Heck, R., Gleicher, M.: Parametric motion graphs. In: Proceedings of the 2007 symposium on interactive 3D graphics and games, pp. 127–139 (2007)
6. Hecker, C., Raabe, B., Enslow, R., DeWeese, J., Maynard, J., Prooijen, K.: Real-time motion retargeting to highly varied user-created morphologies. ACM Trans. Gr. (TOG) **27**(3), 27 (2008)
7. Yoshitaka, A., Ichikawa, T.: A survey on content-based retrieval for multimedia databases. IEEE Trans. Knowl. Data Eng. **11**(1), 81–93 (1999)
8. Forbes, K., Fiume, E.: An efficient search algorithm for motion data using weighted PCA, In: Proceedings of ACM SIGGRAPH/Eurographics symposium on computer animation. pp. 67–76 (2005)
9. Beaudoin, P., Poulin, P., van de Panne, M.: Adapting wavelet compression to human motion capture clips. In: Proceedings of graphics interface, pp. 313–318 (2007)
10. Muller, M., Roder, T., Clausen, M.: Efficient content-based retrieval of motion capture data. ACM Trans. Gr. (TOG) **24**(3), 667–685 (2005)
11. Chen, C., Zhuang, Y., Xiao, J., Liang, Z.: Perceptual 3D pose distance estimation by boosting relational geometric features. Comput. Anim. Virtual Worlds **20**(223), 267–277 (2009)
12. Qi, T., Feng, Y., Xiao, J., Zhuang, Y., Yang, X., Zhang, J.: A semantic feature for human motion retrieval. Comput. Anim. Virtual Worlds **24**(3–4), 399–407 (2013)
13. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching, IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1–8 (2007)
14. Fernando, B., Fromont, E., Muselet, D., Sebban, M.: Supervised learning of Gaussian mixture models for visual vocabulary generation. Pattern Recognit. **45**(2), 897–907 (2012)
15. Zhang, L., Han, Y., Yang, Y., Song, M., Yan, S., Tian, Q.: Discovering discriminative graphlets for aerial image categories recognition. IEEE Trans. Image Process. Publ. IEEE Sig. Process. Soc. **22**(12), 5071–5084 (2013)
16. Xia, Y., Chen, J., Li, J., Zhang, Y.: Geometric discriminative features for aerial image retrieval in social media. Multimedia systems (2014)
17. Zhang, L., Song, M., Zhao, Q., Liu, X., Bu, J., Chen, C.: Probabilistic graphlet transfer for photo cropping. IEEE Trans. Image Process. **22**(2), 802–815 (2013)
18. Zhang, L., Yang, Y., Gao, Y., Yu, Y., Wang, C., Li, X.: A probabilistic associative model for segmenting weakly-supervised images. IEEE Trans. Image Process. **23**(9), 4150–4159 (2014)
19. Zhang, L., Gao, Y., Lu, K., Shen, J., Ji, R.: Representative discovery of structure cues for weakly-supervised image segmentation. IEEE Trans. Multimed. **16**(2), 470–479 (2014)
20. Sun, F., Xu, M., Li, H., Hao, S.: Social video annotation by combining features with a tri-adaptation approach. Multimedia systems (2014)
21. Liu, G., Zhang, J., Wang, W., McMillan, L.: A system for analyzing and indexing human-motion databases. In: Proceedings of ACM SIGMOD international conference on management of data, pp. 924–926 (2005)
22. Safonova, A., Hodgins, J.K., Pollard, N.S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. ACM Trans. Gr. (TOG) **23**(3), 514–521 (2004)
23. Liu, G., McMillan, L.: Segment-based human motion compression, In: Proceedings of ACM SIGGRAPH/Eurographics symposium on computer animation, pp. 127–135 (2006)
24. Gu, Q., Peng, J., Deng, Z.: Compression of human motion capture data using motion pattern indexing. Compu. Gr. Forum **28**(1), 1–12 (2009)
25. Deng, Z., Gu, Q., Li, Q.: Perceptually consistent example-based human motion retrieval, In: Proceedings of the 2009 symposium on interactive 3D graphics and games, pp. 191–198 (2009)
26. Kovar, L., Gleicher, M.: Automated extraction and parameterization of motions in large data sets. ACM Trans. Gr. (TOG) **23**(3), 559–568 (2004)
27. Tang, J.K., Leung, H., Komura, T., Shum, H.P.: Emulating human perception of motion similarity. Comput. Anim. Virtual Worlds **19**(3–4), 211–221 (2008)
28. Tang, J.K., Leung, H.: Retrieval of logically relevant 3D human motions by adaptive feature selection with graded relevance feedback. Pattern Recognit. Lett. **33**(4), 420–430 (2012)
29. Chen, S., Sun, Z., Li, Y., Li, Q.: Partial similarity human motion retrieval based on relative geometry features, In fourth international conference on digital home (ICDH), pp. 298–303 (2012)
30. Yu, T., Shen, X., Li, Q., Geng, W.: Motion retrieval based on movement notation language. Comput. Anim. Virtual Worlds **16**(3–4), 273–282 (2005)
31. Qi, T., Xiao, J., Zhuang, Y., Zhang, H., Yang, X., Zhang, J., Feng, Y.: Real-time motion data annotation via action string. Comput. Anim. Virtual Worlds **25**(3–4), 293–302 (2014)
32. Han, Y., Yang, Y., Ma, Z., Shen, H., Sebe, N., Zhou, X.: Image attribute adaptation. IEEE Trans. Multimed. **16**(4), 1115–1126 (2014)
33. Han Y., Yang Y., Yan Y., Ma Z., Sebe N., Zhou X.: Semi-supervised feature selection via spline regression for video semantic recognition. IEEE Trans. Neural Netw. Learn. Syst. (2014)
34. Müller, M., Röder, T., Clausen, M., Eberhardt B., Krüger, B., Weber, A.: Documentation mocap database HDM05, Technical Reports CG-2007-2, Universität Bonn (2007)
35. Yang, Y., Ma, Z., Hauptmann, G., Sebe, N.: Feature selection for multimedia analysis by sharing information among multiple tasks. IEEE Trans. Multim. **15**(3), 661–669 (2013)